MSBD 5002 Homework 1

Pranav A, 20478966

October 6, 2017

1 Wavelet Transform

Discrete wavelet transform (DWT) is a wavelet transform where the wavelets are discretely sampled.

The expansion series of the wavelet transform is given by:

$$f(t) = \sum_{j,k} a_{j,k}\psi_{j,k}(t) \tag{1}$$

In this formula, the reported values of tuples of discrete wavelet transform is given by $a_{j,k}$ and $\psi_{j,k}$ is the wavelet function. The wavelet function is supported by a scaling function $\varphi(t)$ to scale the coefficient of discrete transform.

For the Haar analysis, this is given by:

$$X = [1, 4, 2, 3, -2, -1, 2, 1]$$

For the first iteration, the given endpoints are $\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\}$.

Since the mapping of endpoints of the rectangular function is also given by $\{0, 1\}$, the scalar coefficients can be directly applied over to the solutions.

Thus we have for the first iteration,

$$X_1 = \left[\frac{5}{\sqrt{2}}, \frac{5}{\sqrt{2}}, \frac{-3}{\sqrt{2}}, \frac{3}{\sqrt{2}}, \frac{-3}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$$

For the second iteration,

$$X_2 = \left[\frac{10}{2}, 0, 0, -\frac{6}{2}, \frac{-3}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$$

For the final iteration,

$$X_3 = \left[\frac{5}{2\sqrt{2}}, \frac{5}{2\sqrt{2}}, 0, -\frac{6}{2}, \frac{-3}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$$

2 Principal Component Analysis

The source code is in the appendix. The given original matrix is:

$$X = \begin{pmatrix} -1 & -1 & 1 \\ -2 & -1 & 4 \\ -3 & -2 & -2 \\ 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 2 & 4 \end{pmatrix}$$

The steps are as follows:

2.1 Normalization

The given matrix would be normalized by making it to a zero mean and unit variance.

$$X = \begin{pmatrix} -0.56287804 & -0.8660254 & -0.2981424 \\ -1.05539632 & -0.8660254 & 1.26710519 \\ -1.5479146 & -1.53960072 & -1.86338998 \\ 0.42215853 & 0.48112522 & -0.2981424 \\ 0.91467681 & 0.48112522 & 0.2236068 \\ 1.40719509 & 1.15470054 & -0.2981424 \\ 0.42215853 & 1.15470054 & 1.26710519 \end{pmatrix}$$

2.2 Covariance Matrix

The formula of covariance is given by:

$$\Sigma_{ij} = \operatorname{cov}(X_i, X_j) = \mathrm{E}\big[(X_i - \mu_i)(X_j - \mu_j)\big] = \mathrm{E}\big[X_i X_j\big] - \mu_i \mu_j$$

Applying this formula to our matrix, we get:

$$\Sigma = \begin{pmatrix} 0.081 & 0.295 & -0.045 & -0.109 & -0.033 & -0.200 & 0.011 \\ 0.295 & 1.663 & -0.237 & -0.554 & -0.373 & -1.185 & 0.390 \\ -0.045 & -0.237 & 0.034 & 0.080 & 0.050 & 0.168 & -0.049 \\ -0.109 & -0.554 & 0.080 & 0.188 & 0.112 & 0.391 & -0.109 \\ -0.033 & -0.373 & 0.050 & 0.112 & 0.122 & 0.277 & -0.155 \\ -0.200 & -1.185 & 0.168 & 0.391 & 0.277 & 0.847 & -0.298 \\ 0.011 & 0.390 & -0.049 & -0.109 & -0.155 & -0.298 & 0.211 \end{pmatrix}$$

2.3 Eigenvalues and Eigenvectors

From the given covariance matrix, the eigenvalues λ and their eigenvectors v would be calculated. Presenting them in the form of augmented matrix:

$$\lambda = \begin{pmatrix} -2.56720630 \times 10^{-16} \\ -7.42799053 \times 10^{-17} \\ -4.34973147 \times 10^{-17} \\ -2.44967436 \times 10^{-17} \\ 2.25595992 \times 10^{-17} \\ 0.185683357 \\ 2.96037113 \end{pmatrix}$$

and their corresponding right eigenvectors:

$$\mathbf{v} = \begin{pmatrix} -0.812 & -0.344 & -0.158 & 0.062 & -0.026 & 0.419 & -0.127 \\ 0.400 & -0.356 & -0.256 & -0.092 & -0.230 & 0.159 & -0.749 \\ -0.144 & 0.253 & -0.338 & -0.825 & -0.326 & -0.064 & 0.106 \\ 0.084 & -0.090 & -0.874 & 0.281 & 0.207 & -0.195 & 0.247 \\ 0.108 & 0.364 & -0.107 & 0.400 & -0.692 & 0.419 & 0.174 \\ 0.215 & -0.737 & 0.130 & -0.130 & -0.301 & 0.024 & 0.535 \\ -0.307 & -0.084 & 0.057 & 0.228 & -0.477 & -0.762 & -0.186 \end{pmatrix}$$

2.4 Proportion of total population variance

The proportion of total population variance of first two components are basically measured by two largest eigenvalues. Thus we would take sum of those two eigenvalues with respect to sum of other eigenvalues.

$$\frac{\lambda_1 + \lambda_2}{\sum_i \lambda_i} = 0.984143852469$$

3 Pattern Discovery

3.1 Discretization

The first step would be to discretize the data.

3.1.1 Categorization

The attributes in the country field would be categorized according to the continents. Thus, the mapping is given by:

1. {USA, Canada} $\mapsto$ North America

ID	Country	Session Length	Web Pages	Buy
1	USA	1213	9	No
2	China	2017	11	Yes
3	Germany	598	35	Yes
4	France	898	45	No
5	Canada	672	9	No
6	Japan	998	14	Yes
7	Korea	1543	18	Yes
8	China	267	7	No
9	USA	1702	13	No
10	England	1345	36	Yes

Table 1: Original Data

2. {China, Japan, Korea} $\mapsto$ Asia
3. {Germany, France, England} $\mapsto$ Europe

ID	Country	Session Length	Web Pages	Buy
1	North America	1213	9	No
2	Asia	2017	11	Yes
3	Europe	598	35	Yes
4	Europe	898	45	No
5	North America	672	9	No
6	Asia	998	14	Yes
7	Asia	1543	18	Yes
8	Asia	267	7	No
9	North America	1702	13	No
10	Europe	1345	36	Yes

Table 2: After categorization

3.1.2 Equal-Depth Binning

The attributes in the session length field would be categorized according to the equal width binning. The interval would be divided into three equi-sized, y simply diving the range by 3. Categorizing it further, we get:

1. $\{267, 598, 672\} \mapsto$ Short
2. $\{898, 998, 1213, 1345\} \mapsto$ Medium
3. $\{1543, 1702, 2017\} \mapsto$ Long

ID	Country	Session Length	Web Pages	Buy
1	North America	Medium	9	No
2	Asia	Long	11	Yes
3	Europe	Short	35	Yes
4	Europe	Medium	45	No
5	North America	Short	9	No
6	Asia	Medium	14	Yes
7	Asia	Long	18	Yes
8	Asia	Short	7	No
9	North America	Long	13	No
10	Europe	Medium	36	Yes

Table 3: After equal width binning

3.1.3 Equal-Depth Binning

The attributes in the web pages field would be categorized according to the equal depth binning. We need to divide that into 4 bins, which can be found using 4 medians.

1. $\{7, 9, 9\} \mapsto$ Less
2. $\{11, 13\} \mapsto$ Medium
3. $\{14, 18\} \mapsto$ High
4. $\{35, 36, 45\} \mapsto$ Very High

ID	Country	Session Length	Web Pages	Buy
1	North America	Medium	Low	No
2	Asia	Long	Medium	Yes
3	Europe	Short	Very High	Yes
4	Europe	Medium	Very High	No
5	North America	Short	Low	No
6	Asia	Medium	High	Yes
7	Asia	Long	High	Yes
8	Asia	Short	Low	No
9	North America	Long	Medium	No
10	Europe	Medium	Very High	Yes

Table 4: After equal depth binning

3.2 Apriori Algorithm

Frequent two itemsets, with a support of at least 3 transactions ($min_{sup} = 0.3$) generated are:

1. Continent = North America, Buy = No
2. Continent = Asia, Buy = Yes
3. Web Pages = Low, Buy = No
4. Continent = Europe, Web Pages = Very High

There are no frequent three itemsets.

3.3 FP Tree

The image of the working solution is attached in supplementary material.

Frequent two itemsets, with a support of at least 3 transactions ($min_{sup} = 0.3$) generated are:

1. Continent = North America, Buy = No
2. Continent = Asia, Buy = Yes
3. Web Pages = Low, Buy = No
4. Continent = Europe, Web Pages = Very High

3.4 Frequent Patterns

Closed frequent patterns are:

1. Continent = North America, Buy = No
2. Continent = Asia, Buy = Yes
3. Web Pages = Low, Buy = No
4. Continent = Europe, Web Pages = Very High

Max frequent patterns are:

1. Continent = North America, Buy = No
2. Continent = Asia, Buy = Yes
3. Web Pages = Low, Buy = No
4. Continent = Europe, Web Pages = Very High

5. Continent = North America
6. Continent = Asia
7. Continent = Europe
8. Session Length = Short
9. Session Length = Medium
10. Session Length = Long
11. Web pages = Low
12. Web pages = Very High
13. Buy = Yes
14. Buy = No

Notes

1. This given solution of the assignment follows the HKUST honour code. Although assignment has been discussed with other peers, the solutions are my own.
2. Kindly give constructive feedback for my incorrect or unclear answers.

Supplementary Material

Listing 1: "Source code of Question 2"

```
import numpy as np
from sklearn.decomposition import PCA
from numpy import linalg as LA

X = np.array([[-1, -1, 1], [-2, -1, 4], [-3, -2, -2],
        [1, 1, 1], [2, 1, 2], [3, 2, 1], [1, 2, 4]]) #Take matrix
normed = (X - X.mean(axis=0)) / X.std(axis=0)
#Normalize the array with zero mean and unit variance
sigma = np.cov(normed)
# Find the covariance. Value of sigma is solution of first part
lambda, v = LA.eigh(sigma)
#Calculate the eigenvalues in lambda and eigenvectors in v,
# eigh reports real values
pca = PCA(n_components=2)
#Initializa the PCA function for 2 components
pca.fit(normed)
```

```
# Apply PCA on to the normalized martix
answer = pca.explained_variance_ratio_
# Retreive the population variances for each components
print(answer[0] + answer[1])
# Final answer is reported by combination of first 2 components
```

Figure 1: Working of FP-Tree